

## APPENDIX

### Interface Specification

#### IDispatch & IErrorInfo

5 All interfaces support IDispatch, which has a built-in IErrorInfo support. This approach will provide an easy way to test the interface, or incorporate into another applications. For the first release, we will not support dual interface.

#### Events

For many occasions, the session object on the server will notify the client with an event that causes the session status to change. These events are defined as a enum here.

```
10 typedef enum NSL_EVENT
    {
        NSLEV                                     = 0,
        NSLEV_PROPERTY_CHANGED,
        NSLEV_UNAUTHORIZED_CHANGED,
15        NSLEV_SESSION_CREATED,
        NSLEV_SESSION_DELETED,
        NSLEV_SESSION_STARTED,
        NSLEV_SESSION_STOPPED,
20        NSLEV_SERVER_ERROR                       = 999
    } NSL_EVENT;
```

| Event                      | Description   |
|----------------------------|---|
| NSLEV_PROPERTY_CHANGED     | A property has been changed. You will get this event when you call CreateSession or RecreateSession as well as changing properties after the session has been created.  |
| NSLEV_UNAUTHORIZED_CHANGED | A property has been changed without the author's knowledge. . You will get this event when the NetShow administrator changes some property of the station, program, or stream that belong to this session on the server. This is potentially dangerous because some properties are coordinated between all components. This may cause it to be out of sync. The session may not operate properly. |
| NSLEV_SESSION_CREATED      | The session has been created. This would be an expected response of CreateSession call, if it has been created successfully. You will not get this event when you call RecreateSession because the session has already been created.  |

|                           |  |
|---------------------------|--|
| NSLEV_SESSION_DELETE<br>D | The session has been deleted. You will get this event when the session has been deleted successfully by call DeleteSession. You will also get this event when a NetShow administrator deletes a station, program or stream that belongs to this session on the server. |
| NSLEV_SESSION_START<br>D  | The session has started. You will get this event when the session has started by calling StartSession. You will also get this event when a NetShow administrator starts the stream that belongs to this session on the server.   |
| NSLEV_SESSION_STOP<br>D   | The session has stopped. You will get this event when the session has started by calling StopSession. You will also get this event when a NetShow administrator stops the stream that belongs to this session on the server.   |
| NSLEV_SERVER_ERROR        | There was an error on the server and the client has lost connection to the server. Most likely, the service has been stopped. You should save the session information and recreate the session later.  |

### IASFSession

| Property | Description  |
|----------|--|
| Control  | Access to the INSLiteSession interface of the object. This is a read-only property.  |
| Property | Access to the IASFSessionProp interface of the object. This is a read-only property. |

| Event         | Description  |
|---------------|--|
| StatusChanged | The session status has changed. The long status parameter identifies the cause of the changes (NSL_EVENT). |

### IASFSessionProp

- 5 All required properties must be set before the session can be created. If you query the property that has not been set (empty), you will get the result of E\_FAIL.

(\*) Required property.

(!) Property cannot be changed after the session is created.

| Property | Description   |
|----------|---|
| Name (!) | The name of the session. The default is null. If specified, it must be unique among the sessions on the server, as well as globally in order to be able to recreate the session later. If not, a globally unique name is generated. |

|  |   |
|--|---|
| Title  | <p>The human-readable title of the session. The default is the same as Name property.</p> <p>Note that the name will appear as 'Show' and 'Clip' title on the client's player.</p>  |
| Description  | <p>The textual description of the session. The default is null.</p> <p>Note that the description will appear on the client's player.</p>  |
| Author   | The author's name. The default is null.   |
| Copyright  | The copyright notice of the content. The default is null.   |
| REX Address (*)  | <p>The IP address or the name of the machine where REX is running. The default is null.</p> <p>This property must be set if you want the server to connect to REX directly. If this property is set, the REX Alias will be ignored.</p> <p>This property must be set if Unicast Only is set. REX Alias cannot be used.</p>            |
| REX Port   | The port on the machine to use to communicate with REX. The default is 7007   |
| REX Alias (* if no REX Address and not using Unicast Only) | <p>The alias that is used to find the REX address. The default is null.</p> <p>This property must be set if you want the server to connect to REX via the alias. If the REX Address is set, this property will be ignored.</p> <p>This property is always ignored, if Unicast Only is set. In this cast, REX address must be set.</p> |
| ASD UNC  | The URL where the ASD file is used to configure REX. The default is null, which means the stream format is one of the SSF. This property is ignored if Unicast Only is set.   |
| Base Directory (*) (!)                                     | Directory path name, in UNC or local file format, where the system can generate and store files that must be accessed by the clients. The NSC and the ASX files required by the session will be created here. The default is null (invalid).  |
| Base URL (*) (!)   | Base URL for the client to access, equivalent of the Base Directory property. The client will access the NSC and ASX files from this base URL. The default is null (invalid).   |
| Client Log URL   | URL that client can use to generate log of its activities and statuses. The actual logging is implemented by a cgi script behind this URL. The default is null (no log created).  |
| Contact Address  | The session's contact address. The default is null.   |
| Contact Phone Number                                       | The session's contact phone number. The default is null.  |
| Contact Email  | The session's contact email. The default is null.   |
| Auto Archive   | Specifies whether the content should be automatically archived.   |

|                        |  |
|------------------------|--|
| Auto Archive Directory | Directory path name where the archive file is generated. Valid only when the Auto Archive property is set. The default is null, which is invalid when the Auto Archive property is set.  |
| Auto Archive Size      | The file size limit of the archive file. The default is 0 (unlimited).   |
| Unicast Only (!)       | The session will be unicasted only. The default is no. If set to yes, all multicast properties will be ignored.  |
| Unicast Rollover (!)   | Specifies whether to allow unicast rollover or not. The default is no. This property is ignored if Unicast Only is set.<br><br>Note that the unicast manager is assumed to be installed on the same machine as the NetShow services.   |
| Multicast Address      | The IP multicast address used for broadcasting. The default is null. If specified, it must be a valid multicast IP address, unique among other addresses used on the server. If not, an address will be generated. This property is ignored if Unicast Only is set.              |
| Multicast Port         | The port used for broadcasting. The default is null. If specified, it must be a valid port, unique on the IP address used on the server. If not, a port number will be generated. This property is ignored if Unicast Only is set.   |
| Multicast TTL          | The multicast time-to-live. The default is 1 (for Intranet). This is the number of 'hops' the multicast packets can make before reaching the destination. This property is ignored if Unicast Only is set.   |
| Drop-Dead Time         | The date and time when the session should already be done. If the session has not been deleted by then, the system will delete it. The default is null (24 hours after it is created). This property makes sure that the server can clean up if for some reason the user didn't. |

### IFTSSession

| Property | Description  |
|----------|--|
| Control  | Access to the INSLiteSession interface of the object. This is a read-only property.  |
| Property | Access to the IASFSessionProp interface of the object. This is a read-only property. |

| Event         | Description   |
|---------------|---|
| StatusChanged | The session status has changed. The long status parameter identifies the kind of changes (NSL_EVENT) that has occurred. |

### IFTSSessionProp

- 5 All required properties must be set before the session can be created. If you query the property that has not been set (empty), you will get the result of E\_FAIL.

(\*) Required property.

(!) Property cannot be changed after the session is created.

| Property  | Description   |
|---|---|
| Name (!)  | The name of the session. The default is null. If specified, it must be unique among the sessions on the server, as well as globally in order to be able to recreate the session later. If not, a globally unique name is generated.   |
| Description                                     | The textual description of the session. The default is null.  |
| Author  | The author's name. The default is null.   |
| Copyright                                       | The copyright notice of the content. The default is null.   |
| Source Base URL (*)                             | The base URL or UNC where the slides are. Wildcards characters are allowed to defines the source file names. The default is null (invalid).<br><br>Note that the files in the subdirectory, if any, will not be transferred.  |
| Output Base URL (*)                             | The base URL that the client will recognize as when the files are finally transferred to the client machine. The source file names are used to complete the URL by concatenating with the base URL. The default is null. This property is used to pre-load the client's URL cache with these files.<br><br>Note that this property must be set if you want the files to go into the URL cache on the client's machine. If this property is set, the Output Base Directory property will be ignored. |
| Output Base Directory (* if no Output Base URL) | The base directory in the client where the files will be transferred to. The default is %TEMP%, which means the files will go into the temporary directory of the client defined by the TEMP environment parameter.<br><br>Note that this parameter is ignored if the Output Base URL property is set.  |
| Redundancy Ratio                                | The percentage of how much data redundancy to be transferred. Using the unreliable transfer protocol, sending redundant data increase the probability that the client would get the data completely. In the intranet, where packet loses are minimal, this can be small. The default is 20%.  |
| Data Bandwidth                                  | The maximum data transfer rate. This is specified in Kbps. The default is 256.  |
| Contact Address                                 | The session's contact address. The default is null.   |
| Contact Phone Number                            | The session's contact phone number. The default is null.  |
| Contact Email                                   | The session's contact email. The default is null.   |
| Multicast Address                               | The IP multicast address used for broadcasting. The default is null. If specified, it must be a valid multicast IP address, unique among other addresses used on the server. If not, an address will be generated.  |
| Multicast Port                                  | The port used for broadcasting. The default is null. If specified, it must be a valid port, unique on the IP address used on the server. If not, a port number will be generated.   |

|                |  |
|----------------|--|
| Multicast TTL  | The multicast time-to-live. The default is 1 (for Intranet). This is the number of 'hops' the multicast packets can make before reaching the destination.  |
| Drop-Dead Time | The date and time when the session should already be done. If the session has not been deleted by then, the system will delete it. The default is null (24 hours after it is created). This property makes sure that the server can clean up if for some reason the user didn't. |

### INSLiteSession

**HRESULT GetServerDateTime (DATE\* pDate)**

|       |   |
|-------|---|
| pDate | The pointer to the date and time, in GMT, returned from the server. |
|-------|---|

This method returns the current server date and time.

Return Value:

|        |  |
|--------|--|
| S_OK   | No error                               |
| E_FAIL | The system fails to provide date/time. |

5 **HRESULT GetStatus (short\* pwStatus)**

|          |                                     |
|----------|-------------------------------------|
| pwStatus | The pointer to the status returned. |
|----------|-------------------------------------|

This method returns the object status. The status may be:

|              |   |
|--------------|---|
| NSL_INACTIVE | The object is instantiated, but not associated with any session on the server.                      |
| NSL_ACTIVE   | The object is associated with a session on the server. The session is not broadcasting any streams. |
| NSL_STARTED  | The object is associated with a session that is in the process of broadcasting the streams          |

Return Value:

|      |          |
|------|----------|
| S_OK | No error |
|------|----------|

**HRESULT CreateSession (void)**

- 10 This method creates the session on the server according to the properties set on the object. The object status will change from NSL\_INACTIVE to NSL\_ACTIVE. Upon successful creation, you will get the NSLEV\_PROPERTY\_CHANGED event and then the NSLEV\_SESSION\_CREATED event.

Return Value:

|                      |  |
|----------------------|--|
| S_OK                 | No error   |
| E_OUTOFMEMORY        | The server is running out of memory.   |
| NS_E_INVALID_REQUEST | The object is already active with a session.                                       |
| E_INVALIDARG         | The properties are not set properly. For example, required properties are not set. |
| NS_E_DUPLICATE_NAME  | The session name is already in use.  |
| WSAADDRESSINUSE      | The multicast IP address and port is already in used.                              |

You may also get other NetShow specific error return value. Please check nerror.h.

#### **HRESULT RecreateSession (BSTR pszName)**

|         |  |
|---------|--|
| pszName | The pointer to the name of the session on the server |
|---------|--|

- 5 This method regains the control and properties of the session that has been created previously by CreateSession method. The object status will change to NSL\_ACTIVE. Previous association with other session will be lost. Upon successful recreation, you will get the NSLEV\_PROPERTY\_CHANGED event.

Return Value:

|              |  |
|--------------|--|
| S_OK         | No error   |
| E_INVALIDARG | The session is not found, or it was not created by this service, or the station, program or stream has been modified beyond recognition by this service. |

- 10 You may also get other NetShow specific error return value. Please check nerror.h.

#### **HRESULT FindSession (BSTR pszName)**

|         |  |
|---------|--|
| pszName | The pointer to the name of the session on the server |
|---------|--|

This method finds the session that has been created previously by CreateSession method.

- 15 Return Value:

|              |   |
|--------------|---|
| S_OK         | No error                                  |
| E_INVALIDARG | The session is not found in the database. |

#### **HRESULT StartSession ((void))**

This method starts broadcasting. The status of the object will change from NSL\_ACTIVE to NSL\_STARTED.

Return Value:

|                      |   |
|----------------------|---|
| S_OK                 | No error  |
| NS_E_INVALID_REQUEST | The object is not active or the session is unicast only. You must call CreateSession or RecreateSession first and create a multicast session. |

- 5 You may also get other NetShow specific error return value. Please check nerror.h.

**HRESULT StopSession (void)**

This method stops broadcasting. The status of the object will change from NSL\_STARTED to NSL\_ACTIVE.

Return Value:

|                      |  |
|----------------------|--|
| S_OK                 | No error   |
| NS_E_INVALID_REQUEST | The object is not active or has already been started. You must call CreateSession or RecreateSession and start it first. |

- 10 You may also get other NetShow specific error return value. Please check nerror.h.

**HRESULT DeleteSession (void)**

- 15 This method deletes the session from the server. If the session is broadcasting, it will stop. The status of the object will change from NSL\_ACTIVE or NSL\_STARTED to NSL\_INACTIVE.

Return Value:

|                      |  |
|----------------------|--|
| S_OK                 | No error   |
| NS_E_INVALID_REQUEST | The object is not ready. You must call CreateSession or RecreateSession first. |

You may also get other NetShow specific error return value. Please check nerror.h.